

第6章分布式系统

虚拟槽分区算法

该算法首先虚拟出一个固定数量的整数集合，该集合中的每个整数称为一个slot槽。这个槽的数量一般是远远大于节点数量的。然后再将所有slot槽平均映射到各个节点之上。例如，Redis分布式系统中共虚拟了16384个slot槽，其范围为[0, 16383]。

而数据只与slot槽有关系，与节点没有直接关系。数据只通过其key的hash(key)映射到slot槽： $slot = hash(key) \% slotNums$ 。这也是该算法的一个优点，解耦了数据与节点，客户端无需维护节点，只需维护与slot槽的关系即可。

Redis数据分区采用的就是该算法。其计算槽点的公式为： $slot = CRC16(key) \% 16384$ 。CRC16()是一种带有校验功能的、具有良好分散功能的、特殊的hash算法函数。

系统搭建与运行

系统搭建

系统架构

下面要搭建的Redis分布式系统由6个节点构成，这6个节点的地址及角色分别如下表所示。一个master配备一个slave，不过master与slave的配对关系，在系统搭建成功后会自动分配。

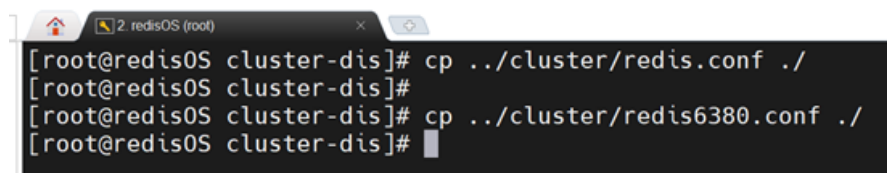
序号	角色	地址
1	master	127.0.0.1:6380
2	master	127.0.0.1:6381
3	master	127.0.0.1:6382
4	slave	127.0.0.1:6383
5	slave	127.0.0.1:6384
6	slave	127.0.0.1:6385

创建目录

在Redis安装目录中mkdir一个新的目录cluster-dis，用作分布式系统的工作目录。

复制2个配置文件

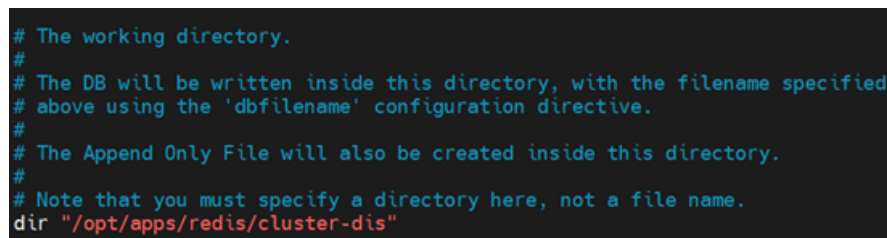
将cluster目录中的redis.conf与redis6380.conf文件复制到cluster-dis目录。



```
[root@redis0S cluster-dis]# cp ../cluster/redis.conf ./
[root@redis0S cluster-dis]#
[root@redis0S cluster-dis]# cp ../cluster/redis6380.conf ./
[root@redis0S cluster-dis]#
```

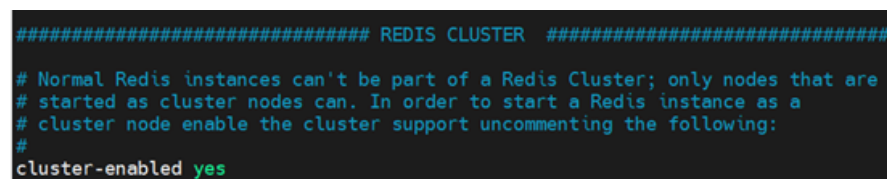
修改redis.conf

对于redis.conf配置文件，主要涉及到以下两个属性：



```
# The working directory.
#
# The DB will be written inside this directory, with the filename specified
# above using the 'dbfilename' configuration directive.
#
# The Append Only File will also be created inside this directory.
#
# Note that you must specify a directory here, not a file name.
dir "/opt/apps/redis/cluster-dis"
```

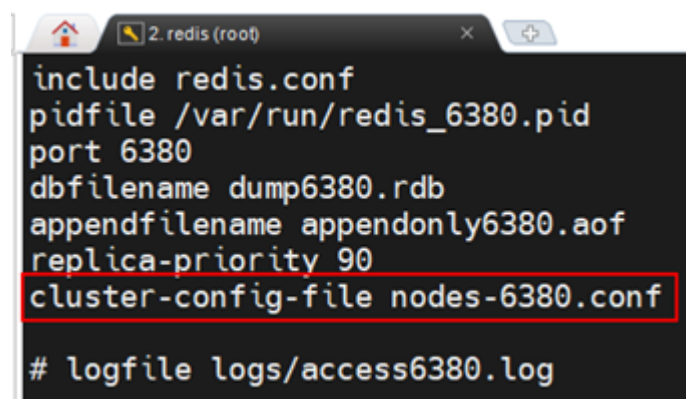
指定工作目录为前面创建的cluster-dis目录。持久化文件、节点配置文件将来都会在工作目录中自动生成。



```
##### REDIS CLUSTER #####
# Normal Redis instances can't be part of a Redis Cluster; only nodes that are
# started as cluster nodes can. In order to start a Redis instance as a
# cluster node enable the cluster support uncommenting the following:
#
cluster-enabled yes
```

该属性用于开启Redis的集群模式。

修改redis6380.conf



```
include redis.conf
pidfile /var/run/redis_6380.pid
port 6380
dbfilename dump6380.rdb
appendfilename appendonly6380.aof
replica-priority 90
cluster-config-file nodes-6380.conf

# logfile logs/access6380.log
```

仅添加一个cluster-config-file属性即可。该属性用于指定“集群节点”的配置文件。该文件会在第一次节点启动时自动生成，其生成的路径是在dir属性指定的工作目录中。在集群节点信息发生变化后（如节点下线、故障转移等），节点会自动将集群状态信息保存到该配置文件中。

复制5个配置文件

使用redis6380.conf复制出5个配置文件redis6381.conf、redis6382.conf、redis6383.conf、redis6384.conf、redis6385.conf。

```
[root@redis0S cluster-dis]# cp redis6380.conf redis6381.conf
[root@redis0S cluster-dis]# cp redis6380.conf redis6382.conf
[root@redis0S cluster-dis]# cp redis6380.conf redis6383.conf
[root@redis0S cluster-dis]# cp redis6380.conf redis6384.conf
[root@redis0S cluster-dis]# cp redis6380.conf redis6385.conf
[root@redis0S cluster-dis]#
```

修改5个配置文件

修改5个配置文件redis6381.conf、redis6382.conf、redis6383.conf、redis6384.conf、redis6385.conf的内容，将其中所有涉及的端口号全部替换为当前文件名称中的端口号。

系统启动与关闭

启动节点

启动所有Redis节点。

```
[root@redis0S cluster-dis]# redis-server redis6380.conf
[root@redis0S cluster-dis]# redis-server redis6381.conf
[root@redis0S cluster-dis]# redis-server redis6382.conf
[root@redis0S cluster-dis]# redis-server redis6383.conf
[root@redis0S cluster-dis]# redis-server redis6384.conf
[root@redis0S cluster-dis]# redis-server redis6385.conf
[root@redis0S cluster-dis]#
```

创建系统

6个节点启动后，它们仍是6个独立的Redis，通过redis-cli --cluster create命令可将6个节点创建了一个分布式系统。

```
[root@redis0S cluster-dis]# redis-cli --cluster create --cluster-replicas 1 192.168.59.110:6380 192.168.59.110:6381 192.168.59.110:6382 192.168.59.110:6383 192.168.59.110:6384 192.168.59.110:6385
```

该命令用于将指定的6个节点连接为一个分布式系统。--cluster replicas 1指定每个master会带有一个slave作为副本。回车后会立即看到如下日志：

```
>>> Performing hash slots allocation on 6 nodes...
Master[0] -> Slots 0 - 5460
Master[1] -> Slots 5461 - 10922
Master[2] -> Slots 10923 - 16383
Adding replica 192.168.59.110:6384 to 192.168.59.110:6380
Adding replica 192.168.59.110:6385 to 192.168.59.110:6381
Adding replica 192.168.59.110:6383 to 192.168.59.110:6382
>>> Trying to optimize slaves allocation for anti-affinity
[WARNING] Some slaves are in the same host as their master
M: 443a8bd88e1520b137901325a4f4ba278a15b130 192.168.59.110:6380
slots:[0-5460] (5461 slots) master
M: 0027f96cf42b5c526549a553205c41bc25c44cdc 192.168.59.110:6381
slots:[5461-10922] (5462 slots) master
M: a130ef41bd3c01f56b91f969b2a4b7bcd84ef52b 192.168.59.110:6382
slots:[10923-16383] (5461 slots) master
S: 4b41d319b5ea63c43125bf2d64b3d03716946706 192.168.59.110:6383
replicates a130ef41bd3c01f56b91f969b2a4b7bcd84ef52b
S: 28e2982f381c458e7e2606b70222ff68e543d1ad 192.168.59.110:6384
replicates 443a8bd88e1520b137901325a4f4ba278a15b130
S: c0a39107b425c60683b5a34b60472ba3d9a9bece 192.168.59.110:6385
replicates 0027f96cf42b5c526549a553205c41bc25c44cdc
Can I set the above configuration? (type 'yes' to accept): yes
```

输入yes后回车，系统就会将以上显示的动态配置信息真正的应用到节点上。

测试系统

```
[root@redis05 cluster-dis]# redis-cli -c -p 6380 cluster nodes
c0a39107b425c60683b5a34b60472ba3d9a9bece 192.168.59.110:6385 slave 0027f96cf42b5c5265
49a553205c41bc25c44cdc 1656936338424 1656936335389 2 connected
28e2982f381c458e7e2606b70222ff68e543d1ad 192.168.59.110:6384 slave 443a8bd88e1520b137
901325a4f4ba278a15b130 0 1656936336000 1 connected
a130ef41bd3c01f56b91f969b2a4b7bcd84ef52b 192.168.59.110:6382 master - 0 1656936336401
3 connected 10923-16383
0027f96cf42b5c526549a553205c41bc25c44cdc 192.168.59.110:6381 master - 0 1656936336000
2 connected 5461-10922
443a8bd88e1520b137901325a4f4ba278a15b130 192.168.59.110:6380 myself,master - 0 165693
6334000 1 connected 0-5460
4b41d319b5ea63c43125bf2d64b3d03716946706 192.168.59.110:6383 slave a130ef41bd3c01f56b
91f969b2a4b7bcd84ef52b 0 1656936337413 3 connected
[root@redis05 cluster-dis]#
```

通过cluster nodes命令可以查看到系统中各节点的关系及连接情况。只要能看到每个节点给出connected，就说明分布式系统已经成功搭建。不过，对于客户端连接命令redis-cli，需要注意两点：

- redis-cli带有-c参数，表示这是要连接一个“集群”，而非是一个节点。
- 端口号可以使用6个中的任意一个。

关闭系统

对于分布式系统的关闭，只需将各个节点shutdown即可。

```
2. redisOS (root)
[root@redis0S cluster-dis]# redis-cli -p 6380 shutdown
[root@redis0S cluster-dis]# redis-cli -p 6381 shutdown
[root@redis0S cluster-dis]# redis-cli -p 6382 shutdown
[root@redis0S cluster-dis]# redis-cli -p 6383 shutdown
[root@redis0S cluster-dis]# redis-cli -p 6384 shutdown
[root@redis0S cluster-dis]# redis-cli -p 6385 shutdown
[root@redis0S cluster-dis]#
```

集群操作

连接集群

无论要怎样操作分布式系统，都需要首先连接上。

```
2. redisOS (root)
[root@redis0S cluster-dis]# redis-cli -c -p 6380
127.0.0.1:6380>
```

与之前单机连接相比的唯一区别就是增加了参数-c。

写入数据

key单个写入

无论value类型为String还是List、Set等集合类型，只要写入时操作的是一个key，那么在分布式系统中就没有问题。

key批量操作

对一次写入多个key的操作，由于多个key会计算出多个slot，多个slot可能会对多个节点。而由于一次只能写入一个节点，所以该操作会报错。

不过，系统也提供了一种对批量key的操作方案，为这些key指定一个统一的group，让这个group作为计算slot的唯一值。

集群查询

查询key的slot

通过cluster keyslot可以查询指定key的slot。例如，下面是查询emp的slot。

```
2. redisOS (root)
192.168.59.110:6382> cluster keyslot emp
(integer) 13178
192.168.59.110:6382>
```

查询slot中key的数量

通过cluster countkeysinslot命令可以查看到指定slot所包含的key的个数。

```
2. redisOS (root)
192.168.59.110:6382> cluster countkeysinslot 13178
(integer) 3
192.168.59.110:6382>
```

查询slot中的key

通过cluster getkeysinslot命令可以查看到指定slot所包含的key。

```
2. redisOS (root)
192.168.59.110:6382> cluster getkeysinslot 13178 3
1) "age{emp}"
2) "depart{emp}"
3) "name{emp}"
192.168.59.110:6382>
```

故障转移

分布式系统中的某个master如果出现宕机，那么其相应的slave就会自动晋升为master。如果原master又重新启动了，那么原master会自动变为新master的slave。